



COURS DE LANGAGE PASCAL

SÉRIE 02

INSTRUCTIONS DE BASE

OBJECTIF PÉDAGOGIQUE :

À l'issue de ce cours, le stagiaire doit être capable d'établir les instructions de bas de langage pascal.

PLAN DE LA LEÇON :

I- INSTRUCTION D'AFFECTATION

II- INSTRUCTION D'ENTRÉE/SORTIE

1- Instruction d'entrée

2- Instruction de sortie

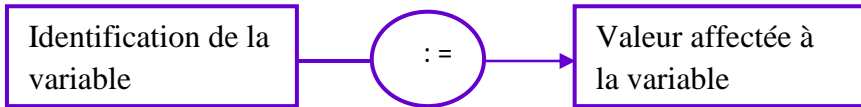
RÉSUMÉ

EXERCICES D'APPLICATION

CORRECTION DES EXERCICES

I- L'INSTRUCTION D'AFFECTATION :

Quand une variable est déclarée, elle est prête à contenir une valeur du même type qu'elle. L'instruction d'affectation ($:=$) est la traduction PASCAL de la Notation Algorithmique (\leftarrow).



Le symbole ' $:=$ ' est encadré à gauche par l'Identificateur de la variable et à droite par la même valeur qui doit lui être effectuée, l'affectation s'effectue de droite à gauche.

EXEMPLE 1 :

PROGRAM Exemple 1 ;

```
    CONST elle = " MERIEME " ;
          Lui = " AISSA " ;
          Nbre = 22 ;
VAR entier : INTEGRER ;    {entier numérique }
    Réel  : REAL ;        { réel numérique }
    Caractère : CHAR ;    {caractère }
    Logique : BOOLEEN;    {variable booléenne }
    a, b, c, d : INTEGRE ; {entiers }
```

BEGIN

```
Entier := nbre ;
Réel := nbre ;
Logique := FALSE ;
Caractère := " T " ;

END.
```

NB : L'affectation $N := N+1$ correspond à une incrémentation de N.

EXEMPLE 2 :

PROGRAM Exemple 2 ;

VAR A, B, C: INTEGER;

BEGIN

A: = 5;

B: = 10;

C: = A +1;

END.

Après la troisième instruction, C vaut 6, A vaut 5 et B vaut 10.

REMARQUE :

- La valeur de la variable réceptrice avant exécution de l'affectation est perdue. (La nouvelle valeur affectée vient écraser l'ancienne).
- La valeur à affecter doit avoir un type identique avec, ou inclus dans le type de la variable réceptrice.

Ainsi après la déclaration

VAR A: REAL

N: INTEGER

L'instruction $A := N+2$ est correcte, le résultat $N+2$ sera converti en réel avant d'être affectée à A. par contre l'affectation $N := A + 0,3$ ne pourra être réalisée (message d'erreur).

II-LES INSTRUCTIONS D'ENTRÉE/SORTIE :

Il n'est question dans cette leçon que des procédures standards des fichiers **INPUT** et **OUTPUT** qui sont associés à des périphériques d'Entrée / Sortie (clavier, écran, imprimantes, etc...).

À ce type de fichiers est associée une notion de ligne en entrée et en sortie à laquelle on ajoute un caractère de fin de ligne.

1- Instruction d'entrée :

Un programme a généralement besoin de données qui ne sont pas toutes connues dans le programme, il s'agit alors de données externes qu'il faut faire connaître à l'ordinateur au moment d'exécution, on parle alors dans ce cas, d'initialisation extérieure de variable.

Ceci pourra être fait par l'exécution d'un ordre de lecture INPUT qui devra être placé en tête du corps du programme.

Il existe deux procédures de lecture READ et READLN qui présentent toutes deux, les mêmes paramètres :

- Le nom du fichier d'entrée (facultatif) par défaut se sera INPUT ;
- La liste des variables (déclarées dans l'instruction VAR) à l'initialiser. Leur type doit être entier, réel ou caractère

a- L'instruction READ :

Si V1, V2 et V3 sont trois variables à initialiser par lecture, on écrira : READ (INPUT, V1, V2, V3) ;

Les variables V1, V2, V3 recevront respectivement comme valeurs trois données successives présentes dans le fichier INPUT.

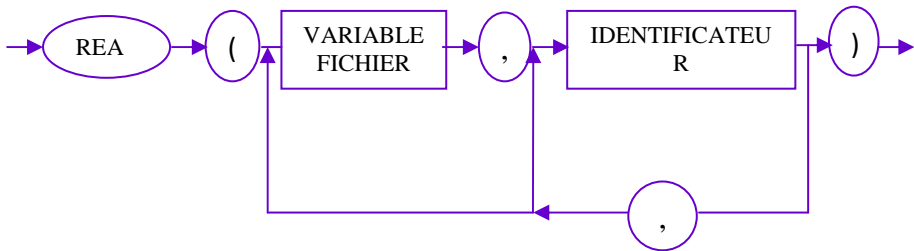
V1 :	1 ^{ère} Valeur	Bien sure chaque donnée doit être du même type que la variable réceptrice
V2 :	2 ^{ème} Valeur	
V3 :	3 ^{ème} Valeur	

Le premier paramètre spécifié peut être une variable de type fichier, c'est-à-dire définissant le support de lecture des données, par défaut se sera le clavier (INPUT). La forme générale de l'instruction READ est écrite en respectant la forme suivante :

READ (Fichier, ident1, ident2,... ident n).

Les identificateurs ident1, ident2,... ident n peuvent être de type différent mais attention à la comptabilité entre les types de données et les variables concernées.

Diagramme de syntaxe de READ



Exemple : Utilisation de READ

Ici, l'entrée se fera par le clavier puisque le paramètre fichier n'est pas mentionné (par défaut c'est le clavier).

```
PROGRAM Exemple 4 ;
  VAR
    a, b, c : INTEGER ;
  BEGIN
    READ (a, b, c);
  END.
```

Les valeurs saisies sont lues en séquences et doivent être séparées par au moins un espace.

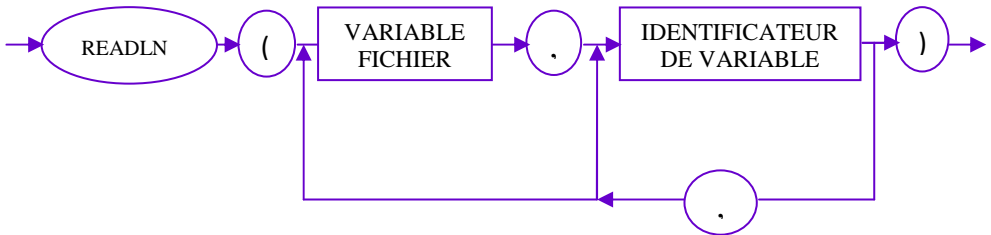
NB : Si une des données lues n'est pas du même type que celui de variables, la compilation est interrompue et un message d'erreurs sera affiché par le compilateur PASCAL.

b- L'instruction READLN :

La syntaxe de READLN est identique à celle de READ. Lorsque le paramètre fichier est omis, le fichier implicitement sélectionné est INPUT.

READLN provoque, de plus un saut à la ligne suivante :

Diagramme de syntaxe de READLN



Exemple : Utilisation de READLN

```
PROGRAM Exemple 5 ;  
  VAR  
    a, b, c : INTEGER ;  
  BEGIN  
    READLN (a, b, c);  
  END.
```

Quelques essais sur la variable A :

Essais	Valeur de A	Résultat obtenu
1	200	200
2	-300	-300
3	32 767	32 767
4	AZ	Erreur de type
5	+35	Erreur le + n'est plus accepté
6	-32 768	-32 768

2- Instruction de sortie :

Un programme étant écrit pour calculer certains résultats, il faut que l'utilisateur puisse en prendre connaissance. Généralement le programme contient des instructions ordonnant à l'ordinateur de les imprimer. Ceci sera fait en écrivant dans le fichier OUTPUT.

En PASCAL, on trouve deux ordres assurant l'impression :

WRITE et WRITELN

Dont les paramètres sont :

- Le nom du fichier de sortie (Facultatif) par défaut, c'est OUTPUT
- La liste des valeurs à écrire. Ces valeurs sont les résultats d'expression, les plus simples étant les constantes et les variables.

a- L'instruction WRITE :

Cette instruction ne conduit pas directement à une impression. Il existe en effet, une « **ligne Tampon** », zone de mémoire spéciale qui est l'image d'une ligne à imprimer et que l'ordinateur remplit au fur et à mesure que l'ordre lui est donné.

EXEMPLE: WRITE (V1, 3*V2, V3)

Pour nous familiariser avec les instructions de visualisation nous allons afficher trois variables lues

```
PROGRAM UNSEPT ;  
    VAR A, B, C: INTEGER;  
BEGIN  
    WRITE (' DONNER 3 ENTREES') ;  
    READLN (A, B, C)  
    WRITE ('A=', A, 'B=', B, 'C=', C);  
END.
```

b- L'instruction WRITELN :

Par rapport à WRITE, cette instruction provoque l'impression immédiate de toute la ligne Tampon, c'est-à-dire sur une même ligne les valeurs déjà présentes sur la ligne Tampon plus les valeurs citées par l'instruction WRITELN que cette ligne soit complète ou non.

Chaque exécution d'un ordre WRITELN imprime une nouvelle ligne ; en particulier l'instruction : WRITELN ; (sans paramètres).

Derrière un ordre WRITELN ordinaire se provoque l'impression d'une ligne blanche (saut de ligne).

REMARQUES :

Il est équivalent d'écrire :

```
WRITELN (OUTPUT, V1, V2, V3);  
WRITE (V1, V2); WRITELN (V3);  
WRITE (V1, V2, V3); WRITELN;
```

Par contre WRITELN (V1) ; WRITELN (V2) ; WRITELN (V3)
conduit à l'impression de 3 valeurs sur 3 lignes différentes.

Il est possible d'indiquer le type du fichier de sortie en utilisant :

- OUTPUT pour l'écran (option par défaut) ;
- LST pour l'imprimante.

EXEMPLE :

```
PROGRAM DEUX6 ;  
{Partie déclarations}  
  
BEGIN  
WRITELN (LST, ' PREMIERE AFFECTATION' ) ;  
WRITELN (LST, ' ET UTILISATION DES INSTRUCTIONS' ) ;  
WRITELN (LST, ' WRITE et WRITELN' ) ;  
WRITELN (LST) ;  
WRITELN (LST, ' VARIABLE A = '365' ) ;  
END.
```



```
PROGRAM DEU X7 ;  
{Partie déclarations vide}
```

```
BEGIN
```

```
    WRITE ('PREMIERE') ;  
    WRITELN (' AFFECTATION') ;  
    WRITE ('et UTILISATION ') ;  
    WRITELN ('des INSTRUCTIONS') ;  
    WRITELN (' WRITE et WRITELN') ;  
    WRITELN ;  
    WRITE (' VARIABLE A = ',365) ;
```

```
END.
```

DIAGRAMME DE SYNTAXE WRITE

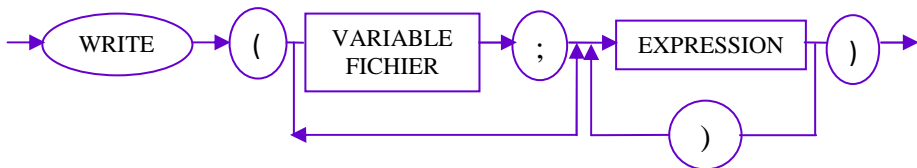
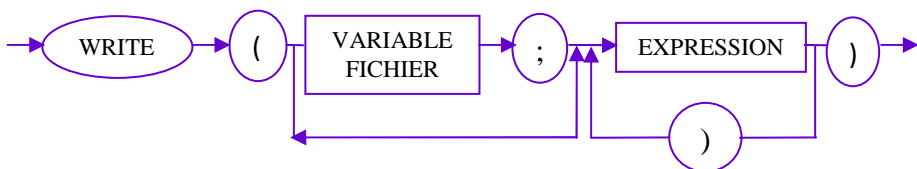


DIAGRAMME DE SYNTAXE WRITELN



RÉSUMÉ :

Dans cette leçon, nous avons vu l'affectation en PASCAL qui ne diffère pas de celle de l'algorithme.

Notons que la variable sur laquelle s'effectue l'affectation et la valeur ou l'expression à affecter doivent avoir le même type.

Ensuite, nous avons abordé les opérations de lecture et d'écriture dans un programme PASCAL.

La lecture se fait directement à partir du clavier au moyen des deux instructions `READ` et `READLN`. L'écriture peut se faire sur écran (c'est-à-dire affichage), sur imprimante ou dans un fichier de données que nous allons voir plus loin au moyen de `WRITE` et `WRITELN`.

EXERCICES D'APPLICATION :

EXERCICE 01 :

Soient x et y deux variables réelles. Ecrire les instructions permettant d'en échanger les valeurs.

EXERCICE 02 :

Ecrire un programme permettant de calculer la surface d'un cercle dont on fournit le rayon (Notez que, en turbo PASCAL, il existe une constante prédéfinie PI contenant la valeur de π).

EXERCICE 03 :

Que produit l'exécution de ce programme :

```
PROGRAM Test 1 ;
VAR n: INTEGER;
    c: CHAR;
    x : REAL ;
BEGIN
n := 10 ; c : 'A' ; x := 1,5 ;
READLN (n, c, x);
WRITE (n); Writeln; Writeln (x);
END.
```

EXERCICE 04 :

Qu'affiche le programme ci-dessus

```
PROGRAM Affiche ;
{Déclarations vide }
BEGIN
    WRITE ('Première affectation') ;
    Writeln (' et utilisation des instructions') ;
    Writeln (' write et writeln') ;
    Writeln (' WRITE et Writeln') ;
    Writeln ;
    Writeln (' VARIABLE A = 365') ;
END.
```

EXERCICE 05 :

Exécuter le programme suivant et afficher le résultat.

```
PROGRAM suivant ;  
VAR a, b, c ! Integer ;  
  
BEGIN  
    WRITE ('entre 3 entiers séparés par des espaces ==> : ');  
    READ (a, b, c) ;  
    WRITELN ;  
    WRITELN ('A=' a,'B=', b, 'C=', c)  
END.
```

EXERCICE 06 :

Donner et commenter le résultat de 2 programmes :

PROGRAM EX1 ;	PROGRAM EX2 ;
VAR A,B,C : INTEGER	VAR A,B,C : INTEGER
BEGIN	BEGIN
A := 5 ;	READLN (A ,B) ;
B := 6 ;	C := B + A
C := A + B ;	WRITELN (' le résultat' ; c) ;
Writeln (' le résultat =' ; c) ;	c := B + C
C := A + B	Writeln (' le résultat' ; c) ;
Writeln (' le résultat ; c) ;	END.
END.	

CORRECTION DES EXERCICES :

EXERCICE N° 01 :

```
PROGRAM Echange ;  
VAR x, y, z: REAL ;  
BEGIN  
  x := 4 ;  
  y := 6 ;
```

```
  WRITELN (' avant l'échange') ;  
  WRITELN (' x =', x) ;  
  WRITELN (' y =', y) ;  
  Z := x ;  
  x := y ;  
  y := z ;  
  WRITELN (' après échange') ;  
  WRITELN (' x =', x) ;  
  WRITELN (' y =', y) ;  
  AND.
```

Exécution
avant l'échange
x = 4
y = 6
après l'échange
x = 6
y = 4

EXERCICE N° 02 :

```
PROGRAM surface ;  
VAR rayon, surface : REAL ;  
BEGIN  
  WRITE (' donner le rayon= ') ;  
  READLN (rayon) ;  
  Surface :=, PI*sqr (rayon) ;  
  
  Writeln ('surface =', surface) ;  
  
END.
```

Exécution

Donner le rayon : 4
Surface = 40.24

EXERCICE 03 :

v25 V 2.77
25V

En premier lieu on affecte les valeurs suivantes aux variables déclarées préalablement

n := 10 ; c := 'A' ; X := 1,5 ;
mais ces valeurs seront vite remplacées par :

n := 25
c := 'V' → par l'instruction Readln (n, c, x)
x := 2,75

à la fin on affiche ces valeurs de la manière suivante :

25 V par l'instruction write (n) et writeln ©
2,77 par l'instruction writeln (x)

EXERCICE 04 :

Le programme affiche les lignes suivantes :
Première affectation et l'utilisation des instructions write et writeln

Variable A = 365

EXERCICE 05 :

Affichage des résultats
Entrez 3 entiers séparés par des espaces == => : 4 5 8

A = 4 B = 5 C = 8

EXERCICE 06 :

Le résultat du programme EX 1 : On déclare trois variables entières dont on leur affecte les valeurs : (A := 5 et B := 6). Puis on effectue l'addition A + B et on affiche le résultat.

Une autre opération de multiplication et effectuer (A*B) et enfin on affiche le résultat

Résultat = 15
Résultat = 30

Même type de programme pour EX2 sauf qu'ici, on n'affecte pas initialement les valeurs aux variables A et B. On les lit à partir du clavier la suite est similaires au programme EX1, on prend les mêmes valeurs : 5 6

Résultat = 15

Résultat = 30